Applied Soft Computing Journal 98 (2021) 106735

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

# Performance analysis of nonlinear activated zeroing neural networks for time-varying matrix pseudoinversion with application

Zeshan Hu [a], Lin Xiao [b,*], Kenli Li [a,*], Keqin Li [a], Jichun Li [c]

[a] College of Information Science and Engineering, Hunan University, Changsha 410082, China
[b] Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, 410081, China
[c] School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

ARTICLE INFO

*Article history:*
Received 11 March 2020
Received in revised form 3 September 2020
Accepted 13 September 2020
Available online 22 September 2020

*Keywords:*
Time-varying matrix pseudoinverse
Finite-time
Nonlinear activation functions
Zeroing neural network
Robot manipulator

ABSTRACT

By exploiting two simplified nonlinear activation functions, two zeroing neural network (ZNN) models are designed and studied to efficiently tackle the time-varying matrix pseudoinversion problem. Compared with ZNN activated by previously presented activation functions, these two simplified finite-time ZNN (SFTZNN) models (called SFTZNN1 and SFTZNN2) not only achieve faster finite-time convergence, but also possess better robustness. In addition, the SFTZNN1 and SFTZNN2 models have simpler structure compared with the widely used sign-bi-power activated ZNN model. Theoretical analysis is presented to obtain the maximum convergence time for the SFTZNN models in ideal conditions. Besides, when external perturbations are injected into the proposed SFTZNN models, upper bounds of the steady-state residual error are theoretically calculated. Comparative simulations and one engineering application case validate the feasibility and superiority of the two new SFTZNN models when solving time-varying matrix pseudoinversion.



## 1. Introduction

Finding the Moore–Penrose pseudoinverse of a matrix has always been an essential problem in various science and engineering fields, including signal processing [1], image noise reduction [2], tensor nearness problem [3], adaptive robust control [4] and robotics [5]. Due to the variety of applications of matrix pseudoinverse, many researchers have investigated numerical algorithms for this problem. For example, Newton's iteration method was used in [6] and [7] to obtain the Moore–Penrose pseudoinverse while Grevilles recursive method was investigated in [8] and singular value decomposition was studied in [9]. More progress such as using Cholesky factorization algorithm to compute the matrix pseudoinverse has been made for addressing this problem. However, all of these numerical algorithms possess serial-processing property. In addition, the minimum time complexity of these algorithms is no less than the cube of the matrix dimensions. Therefore, these numerical algorithms usually are unable to meet the requirements to delivery large scale computation within very limited time. For example, when applying these algorithms to online or real-time applications, we need to execute them at every sampling period and increase the sampling rate for better accuracy. However, these conventional numerical algorithms cannot complete such tasks due to time cost that is larger than the limited running time gap.

Recently, due to the comprehensive study on neural networks [10–12], recurrent neural network (RNN) has been extensively investigated by researchers [13–17]. RNN models can be implemented physically by hardware, and their computation process can be parallel and distributed, which means high computation efficiency. Thus, neural network methods usually have less time complexity than conventional numerical methods. Moreover, because of its capability to solve numerous mathematical time-varying problems, continuous-time RNN model is considered to be an efficient tool to deal with online problems like the time-varying matrix pseudoinversion. Traditional gradient-based neural networks (GNNs) use scalar-type error value to monitor the performance. The neural network will make the error norm converge to zero with the gradient-descent method [18,19]. However, GNNs do not perform well in solving the time-varying problems with huge lagging-errors. Then a special recurrent network called the zeroing neural network (ZNN) was proposed in [20,21] to deal with such problem. By utilizing the time-derivative information of dynamic coefficients, ZNN models successfully tackle time-varying problems via eliminating the lagging-error. Generally, ZNN models use an indefinite function called error function to monitor the error. By designing different activation functions, new ZNN models for the same problem can be greatly enriched with different performance. Furthermore, by

* Corresponding authors.
*E-mail addresses:* xiaolin860728@163.com (L. Xiao), lkl@hnu.edu.cn (K. Li).

https://doi.org/10.1016/j.asoc.2020.106735
1568-4946/

applying a special activation function called the sign-bi-power (Sbp) function and some other novel activation functions, ZNN models are able to solve a time-varying problem in finite time [22–26].

Following the inspiration of accelerating ZNN models by using finite-time convergent activation functions, in this paper, two nonlinear activation functions are explored to establish two simplified finite-time ZNN (SFTZNN) models to solve the time-varying pseudoinverse matrices. Generally, ZNN models are assumed to be working in condition free of any type of perturbation. However, perturbations are unavoidable in real world. Perturbations may be caused by realization errors and other external errors, substantially reducing the accuracy of ZNN models. For this reason, we further investigate the robustness of two SFTZNN models and compare the results with other ZNN models. For better illustrating the highlights of this paper, we compare our work with some recent results (i.e., [27–30]), and the comparative results have been listed in Table 1. From this table, it can be concluded that our work is not only efficient on solving the time-varying/static pseudoinversion problems, but also has faster finite-time convergence speed. Besides, we compare our work with Ref. [31] in Tables 2 and 3, which show the main contributions and novelties from our work. From the comparison of these results, it can be concluded that our work has made major contributions in terms of time-varying problem solving, model design, activation function comparison, rigorous robustness analysis, and robotic motion application. In addition, unlike other works that ignore robustness analysis or only study single type of noise, our work investigates the robustness performance of the proposed model in the presence of different types of noises. The following is a summary of the major contributions of this paper.

(1) Two simplified finite-time convergent activation functions are introduced and studied for time-varying matrix pseudoinverse problem solving, because they have less complexity than the Sbp function.
(2) Two SFTZNN models are developed for obtaining the pseudoinverse of a time-varying matrix based on these two simplified activation functions. Furthermore, the upper bounds of convergence time are provided.
(3) Perturbed SFTZNN models are proposed to illustrate the superior robustness of two simplified activation functions, with theoretical proofs given to determine the steady-state residual error upper bounds under perturbations.
(4) The comparative numerical experiments demonstrate better convergence performance of two SFTZNN models in the presence of various perturbations. In addition, an application on redundant manipulators further illustrates the feasibility of SFTZNN models in real world.

## 2. Problem description and ZNN model

This section contains two parts. First, let us present the mathematical description of the time-varying matrix pseudoinverse. Then, we introduce and generalize a ZNN model for such a problem.

### 2.1. Problem description

Pseudoinverse is also called the Moore–Penrose inverse. In mathematics, given a dynamic matrix $M(t) \in \mathbb{R}^{m\times n}$, the pseudoinverse $M^{+}(t) \in \mathbb{R}^{n\times m}$ is a matrix that satisfies the following equations [32]:

$$\begin{cases} M^{+}(t)M(t)M^{+}(t) = M^{+}(t), \\ M(t)M^{+}(t)M(t) = M(t), \\ \big(M(t)M^{+}(t)\big)^{\mathrm{T}} = M(t)M^{+}(t), \\ \big(M^{+}(t)M(t)\big)^{\mathrm{T}} = M^{+}(t)M(t). \end{cases}$$

It should be noted that pseudoinverse $M^{+}(t)$ exists for any matrix $M(t)$ [32]. Particularly, if $M(t)$ is full rank all the time, i.e., $\mathrm{rank}(M(t)) = \min\{m, n\}$, $\forall t \in [0, +\infty)$, then $M^{\mathrm{T}}(t)M(t)$ is nonsingular when $m > n$ while $M(t)M^{\mathrm{T}}(t)$ is nonsingular when $m < n$. Thus we use following equations to obtain the dynamic pseudoinverse of $M(t)$ [32]:

$$M^{+}(t) = \begin{cases} M^{\mathrm{T}}(t)\big(M(t)M^{\mathrm{T}}(t)\big)^{-1}, & \text{if} \quad m \leqslant n, \\ \big(M^{\mathrm{T}}(t)M(t)\big)^{-1}M^{\mathrm{T}}(t), & \text{else.} \end{cases} \tag{1}$$

In this paper, we assume that the first order time derivative of $M(t)$ is continuous and $M(t)$ is a full rank matrix all the time, which guarantees the existence and uniqueness of solution $M^{+}(t)$. Furthermore, we only consider the condition that $m > n$, because the situation of $m \leqslant n$ is similar to the case of $m > n$.

To lay foundations for theoretical analysis, we introduce the following Lemma to guarantee the boundness of $(M^{\mathrm{T}}(t)M(t))^{-1}$ [20].

**Lemma 1.** *There exists a number $\alpha > 0, \alpha \in \mathbb{R}$ that*

$$\min_{\forall i\in\{1,\ldots,n\}} \big|\lambda_i(M^{\mathrm{T}}(t)M(t))\big| \geqslant \alpha, \quad \forall t \geqslant 0. \tag{2}$$

*where $\lambda_i(\cdot)$ denotes the ith eigenvalue of matrix $M^{\mathrm{T}}(t)M(t) \in \mathbb{R}^{n\times n}$. If the norm of $M^{\mathrm{T}}(t)M(t)$ satisfies $\big\|M^{\mathrm{T}}(t)M(t)\big\|_{\mathrm{F}} \leqslant \varepsilon_M$, $\forall t \in [0, +\infty)$, then $\|(M^{\mathrm{T}}(t)M(t))^{-1}\|_{\mathrm{F}}$ is also uniformly upper bounded by a scalar [20]:*

$$\varphi(\alpha, \varepsilon_M^2, n) = \sum_{i=0}^{n-2} \frac{C_n^i \varepsilon_M^{2(n-i-1)}}{\alpha^{n-i}} + \frac{n^{3/2}}{\alpha}.$$

*where $C_n^i := n!/(i!(n-i)!)$, $\varepsilon_M \in \mathbb{R}$ is the upper bound of the Frobenius norm of $M^{\mathrm{T}}(t)M(t)$.*

### 2.2. ZNN

ZNN has been developed and introduced first by Zhang et al. since 2002. Then, in the past decades, ZNNs were extensively developed to deal with different time-varying problems, including the time-varying matrix pseudoinverse. Compared with GNN, ZNN possesses higher convergence speed, and eliminates the large steady-state residual error that appears in the former one. The design process of ZNN for matrix pseudoinverse problem is illustrated below [27,33].

We first define an error matrix $E(X(t), t) \in \mathbb{R}^{n\times m}$ rather than the scalar energy function to evaluate the process of matrix pseudoinversion, which is presented in the following:

$$E(X(t), t) := M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t), \tag{3}$$

where $M(t) \in \mathbb{R}^{m\times n}$ is a dynamic full rank matrix; and, dynamic matrix $X(t) \in \mathbb{R}^{n\times m}$ representing the pseudoinverse of $M(t)$ is unknown and should be obtained.

Then, error matrix's derivative $\dot{E}(X(t), t)$ is designed to force each element $e_{ij}(t), i = \{1, \ldots, n\}; j = \{1, \ldots, m\}$ of $E(X(t), t)$ converge to zero as time goes. Based on this idea, $\dot{E}(X(t), t)$ can be formulated particularly as follows:

$$\frac{dE(X(t), t)}{dt} = -\Gamma F(E(X(t), t)), \tag{4}$$

where $F(\cdot) : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ represents a matrix mapping, whose elements are the same and denoted by $f(\cdot)$. For the clarity and consistence of analysis, in this paper we assume $\Gamma = \gamma I$, where $\gamma > 0 \in \mathbb{R}$ is a design parameter that can adjust the convergence speed of ZNN. Combining the formula (4) with another formula (3), we get the dynamic equation of the ZNN model for matrix pseudoinversion:

$$\begin{aligned} M^{\mathrm{T}}(t)M(t)\dot{X}(t) = &-\big(\dot{M}^{\mathrm{T}}(t)M(t) + M^{\mathrm{T}}(t)\dot{M}(t)\big)X(t) \\ &+ \dot{M}^{\mathrm{T}}(t) - \gamma F\big(M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)\big), \end{aligned} \tag{5}$$

**Table 1**
The main novelties and differences of this work from Refs. [27–30].

| # | Item | [27] | [28] | [29] | [30] | Our paper |
|---|---|---|---|---|---|---|
| 1 | Problem type | Static | Time-varying | Time-varying | Time-varying | Time-varying |
| 2 | Activation function | Linear | sign-bi-power | sign-bi-power | Linear | SFTAF2 |
| 3 | Robustness analysis | No | No | No | Yes | Yes |
| 4 | Noise type | None | None | None | Model | Model & differential |
| 5 | Convergence | Exponentially | Finitely | Finitely | Exponentially | Finitely (accelerated) |

**Table 2**
The main novelties and differences of this work from Ref. [31].

| # | Problem | Type | Model | AF comparisons | Application |
|---|---|---|---|---|---|
| Our work | Pseudoinverse | Time-varying | ZNN | Abundant | Robot manipulator control |
| [31] | Quadratic programming | Static | DNN | None | Parameters estimation |

**Table 3**
The main novelties and differences of this work from Ref. [31].

| # | Quantitative robustness analysis | Noise elements | Noise type (analytical) | Noise type (experimental) |
|---|---|---|---|---|
| Our work | Rigorous | Differential & model | Unknown & time-varying | Sinusoidal |
| [31] | None | Model | None | Mean Gaussian White |

where $X(t)$ is the state matrix as well as the output of neural network, with initial state being $X(0) \in \mathbb{R}^{n\times m}$.

Thanks to previous research, we have already known that the convergence speed and the robustness of ZNNs varies when using different functions $f(\cdot)$. In general, activation functions of ZNNs can be divided into the following:

(1) Linear activation function

$$f(x) = x, \tag{6}$$

(2) Power activation function

$$f(x) = x^p, \quad p \geqslant 3, \tag{7}$$

(3) Bipolar Sigmoid activation function

$$f(x) = \frac{1-\exp(-\xi x)}{1+\exp(-\xi x)}, \quad \xi > 2, \tag{8}$$

(4) Power-Sigmoid activation function

$$f(x) = \begin{cases} x^q & \text{if} \quad |x| \geqslant 1, \\ \frac{(1+\exp(-\xi))}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi x)}{1+\exp(-\xi x)} & \text{else.} \end{cases} \tag{9}$$

where $\xi > 2$ and $q \geqslant 3$.

It has been known that when the linear activation function is applied, ZNNs possess exponential convergence rate. But it should also be noted that, due to the nature of the exponential function, the closer the error function $e_{ij}(t)$ gets to zero, the slower the convergence rate is. To overcome this shortcoming, researchers introduce nonlinear activation functions to accelerate the convergence speed of ZNNs. Activation functions (7), (8) and (9) are all potential choices that can be used to achieve this goal. In-depth study shows that (7), (8) and (9) still cannot achieve finite-time convergence, which means that each $e_{ij}(t)$ does not decrease to zero after infinite time. Then, the Sbp activation function was introduced to accomplish finite-time convergence [22,23], which is presented as below:

$$f(x) = \frac{1}{2}\text{sgn}^{\tau}(x) + \frac{1}{2}\text{sgn}^{\frac{1}{\tau}}(x), \quad \tau \in (0, 1), \tag{10}$$

with $\text{sgn}^{\tau}(x)$ defined as

$$\text{sgn}^{\tau}(x) = \begin{cases} |x|^{\tau} & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x = 0, \\ -|x|^{\tau} & \text{if} \quad x < 0. \end{cases}$$

Recently, two new activation functions modified from the Sbp function have been proposed in [31,34], which are called the Simplified Finite-Time Activation Function 1 (SFTAF1):

$$f(x) = \text{sgn}^{\tau}(x), \quad \tau \in (0, 1), \tag{11}$$

and the Simplified Finite-Time Activation Function 2 (SFTAF2):

$$f(x) = \beta_1\text{sgn}^{\tau}(x) + \beta_2 x, \quad \tau \in (0, 1), \tag{12}$$

where $\beta_1 > 0$ and $\beta_2 > 0$. For brief expression, in the following discussions, ZNN model (5) will be named as the Simplified Finite-Time Zeroing Neural Network 1 (SFTZNN1) model or the Simplified Finite-Time Zeroing Neural Network 2 (SFTZNN2) model when using the activation function SFTAF1 or SFTAF2. The above two SFTAFs differ from other activation functions on multiple aspects. As can be seen from the SFTAF1 (11) and the SFTAF2 (12), when compared with the linear, power, bipolar and power-sigmoid activation functions, the special piecewise $\text{sgn}^{\text{fi}}(x)$ function have been added which gives them finite-time convergence property. When compared with the Sbp activation function, the SFTAF1 and the SFTAF2 have dropped the power function part: $\text{sgn}^{1/\tau}(x)/2$, which makes their structure simpler and their computation complexity lower.

To the most of our knowledge, there has not been enough efforts that use the SFTAF1 or the SFTAF2 on solving the time-varying matrix pseudoinversion yet. Furthermore, the robustness of the proposed two SFTZNN models has not been well studied under external perturbation.

## 3. Convergence analysis

Section 2 gives a general ZNN model that can be used to obtain the time-varying matrix pseudoinversion as well as several kinds of activation functions, such as the SFTAF1 and the SFTAF2. In this section, we will theoretically analyze and compare the convergence speed between different activation functions, in which the superiorities of SFTAF1 and SFTAF2 will be demonstrated.

**Proposition 1.** *Given a time-varying matrix $M(t) \in \mathbb{R}^{m\times n}$ satisfying Lemma 1, if monotonically increasing odd activation function $f(\cdot)$ is used, then the state matrix $M(t)$ of ZNN model (5) starting from arbitrary initial states $M(0) = M_0 \in \mathbb{R}^{n\times m}$ will always converge to the theoretical pseudoinverse $M^*(t)$ of time-varying matrix $M(t)$.*

**Theorem 1.** *In the condition of Proposition 1, if SFTAF1 (11) is used, the SFTZNN1 model will converge to the theoretical pseudoinverse $M^{+}(t)$ globally in finite time*

$$t_f \leqslant \frac{|E_{\max}(0)|^{1-\tau}}{\gamma(1-\tau)}, \tag{13}$$

*where $E_{ij}(t)$ represents the ijth element in $E(t)$, and $|E_{\max}(0)| = \max\{|E_{ij}(0)|\}$.*

**Proof.** It can be referred in Refs. [22,31,34,35], and thus omitted.

**Theorem 2.** *In the condition of Proposition 1, if SFTAF2 (12) is used, the SFTZNN2 model will converge to the theoretical pseudoinverse $M^{+}(t)$ globally in finite time*

$$t_f \leqslant t_{\max} = \frac{1}{\rho_2(1-\tau)} \ln \frac{\rho_2|E_{\max}(0)|^{1-\tau} + \rho_1}{\rho_1}, \tag{14}$$

*where $E_{\max}(0)$ is defined the same as in Theorem 1, $\rho_1 = \gamma\beta_1$ and $\rho_2 = \gamma\beta_2$.*

**Proof.** It can be referred in Refs. [22,31,34], and thus omitted.

**Proposition 2.** *Under the given conditions of Proposition 1, if we use the linear, bipolar-sigmoid, power, and power-sigmoid activation functions in Section 2, then the neural network is unable to make $e_{ij}(t)$ converge to zero in finite time.*

**Proof.** Let us respectively consider the situations of using the linear, bipolar-sigmoid, power, and power-sigmoid activation functions.

(1) For the simple linear function, the equality $\dot{e}_{ij}(t) = -\gamma e_{ij}(t)$ exists, and the entry error is $e_{ij}(t) = \exp(-\gamma t)e_{ij}(0)$, meaning that $|e_{ij}(t)| = |\exp(-\gamma t)e_{ij}(0)| > 0$ with $t \in (0, +\infty)$. Obviously in finite time, entry error $e_{ij}(t)$ will not converge to zero.

(2) For the bipolar-sigmoid case, we first assume that it is able to achieve finite-time convergence within interval $[0, t_1]$ where $t_1 \in \mathbb{R}^{+}$ is the finite convergence time upper bound. Then, ZNN model (5) can be written as an equivalent differential equation:

$$\frac{\mathrm{d}e_{ij}(t)}{\mathrm{d}t} = -\gamma \cdot \frac{1 - e^{-\xi e_{ij}(t)}}{1 + e^{-\xi e_{ij}(t)}}.$$

Such an equality can be rewritten as

$$\gamma \mathrm{d}t = -\frac{1 + e^{-\xi e_{ij}(t)}}{1 - e^{-\xi e_{ij}(t)}} \mathrm{d}e_{ij}(t).$$

Then, integrating the above equality from initial time to time instant $t_1$:

$$\int_0^{t_1} \gamma \mathrm{d}t = \int_{e_{ij}(0)}^{e_{ij}(t_1)} -\frac{1 + e^{-\xi e_{ij}(t)}}{1 - e^{-\xi e_{ij}(t)}} \mathrm{d}e_{ij}(t),$$

$$t_1 = \frac{1}{\gamma} e_{ij}(0) - \frac{2}{\xi} [\ln|1 - e^{-\xi e_{ij}(t)}|]_{e_{ij}(0)}^{e_{ij}(t_1)}.$$

Because $\lim_{e_{ij}(t_1)\to 0} \ln|1 - e^{-\xi e_{ij}(t_1)}| = -\infty$, so $\lim_{e_{ij}(t_1)\to 0} t_1 = +\infty$, which contradicts with our previous assumption for $t_1$. Thus the bipolar-sigmoid activation function cannot converge in finite time.

(3) Consider the power function $f(x) = x^p$, ZNN model (5) becomes $\dot{e}_{ij} = -\gamma e_{ij}^p$, the solution can be derived as $e_{ij}(t) = e_{ij}(0)\{(p-1)e_{ij}^{p-1}(0)\gamma t + 1\}^{-1/(p-1)}$. Clearly, $|e_{ij}(t)| > 0$ holds for arbitrary $t \geqslant 0$. Therefore, we can conclude that the power function is not able to achieve finite-time convergence.

(4) For the power-sigmoid function, this is a piecewise function. In the evolvement of the neural network, we suppose that $t_f$ is the finite time instant when $e_{ij}(t)$ converges to zero. There must exist $\delta > 0$ making $|e_{ij}(t)|$ fall into $(0, 1)$ in the time span $(t_f - \delta, t_f)$. However, from the above proof associated with the bipolar-sigmoid case, we conclude that the part of activation function (9) defined in $x \in (0, 1)$ cannot achieve finite-time convergence. This contradiction proves that the power-sigmoid function cannot converge in finite time.

In a word, the neural network is unable to make $e_{ij}(t)$ converge to zero in finite time when using the linear, bipolar-sigmoid, power, and power-sigmoid activation functions.

**Remark 1.** From Theorems 1–2 and Proposition 2, when using activation functions (6)–(9), although the ZNN model can converge to the theoretical solution of the problem, it is not able to achieve finite-time convergence. In addition, Theorems 1–2 prove that, using the new activation functions SFTAF1 and SFTAF2, the entry error $e_{ij}(t)$ of the ZNN model will converge to zero globally and in finite time. Hence, compared to activation functions (6), (7), (8) and (9), the SFTAF1 and SFTAF2 possess superior convergence performance. Furthermore, as proved in [28], the convergence time upper bound of the ZNN model using (10) is $\max\{2|E^{-}(0)|^{1-\tau}/[\gamma(1-\tau)], 2|E^{+}(0)|^{1-\tau}/[\gamma(1-\tau)]\}$, being two times as much as that using the SFTAF1. Evidently, the SFTZNN1 and SFTZNN2 also possess superior convergence performance compared to the ZNN model using the Sbp function. Besides, the SFTAF1 and SFTAF2 has less computation amount than the Sbp function, which makes them more suitable for real world applications.

## 4. Robustness analysis

When implementing neural networks using circuits, some perturbations may exists. For example, components in analog circuits may result in high-order residual errors, while truncation or roundoff errors may happen in digital circuits too. The aforementioned errors eventually may cause model-implementation errors in neural networks [36].

The ZNN model (5) is proposed to find the time-varying matrix pseudoinverse under ideal conditions free of errors. But in realistic applications, the errors aforementioned are unavoidable and should be eliminated or reduced. In order to address this problem, we add the differential error and the model-implementation error into ZNN model (5). Thus we get following dynamics:

$$\begin{aligned} M^{\mathrm{T}}(t)M(t)\dot{X}(t) =& \dot{\hat{M}}^{\mathrm{T}}(t) - \left(\dot{\hat{M}}^{\mathrm{T}}(t)M(t) + M^{\mathrm{T}}(t)\dot{\hat{M}}(t)\right)X(t) \\ & - \gamma F\left(M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)\right) + \Delta_C(t), \end{aligned} \tag{15}$$

where $\Delta_C(t) \in \mathbb{R}^{n\times m}$ denotes the model implementation error, and $\dot{\hat{M}} := \dot{M} + \Delta_B$ represents the time derivative of $M(t)$ with $\Delta_B(t) \in \mathbb{R}^{m\times n}$ denoting the differential error.

If we substitute $\dot{\hat{M}} := \dot{M} + \Delta_B$ into (15), then the ZNN model with perturbation can be reformulated as the following dynamic:

$$\begin{aligned} M^{\mathrm{T}}(t)M(t)\dot{X}(t) =& \dot{M}^{\mathrm{T}}(t) - \big[\ \left(\dot{M}(t) + \Delta_B(t)\right)^{\mathrm{T}}M(t) \\ & + M^{\mathrm{T}}(t)\left(\dot{M}(t) + \Delta_B(t)\right)\ \big]X(t) \\ & - \gamma F\left(M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)\right) \\ & + \Delta_B^{\mathrm{T}}(t) + \Delta_C(t). \end{aligned} \tag{16}$$

Then, defining error matrix $E(t) = M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)$, dynamics (16) can be rewritten as

$$\begin{aligned} \dot{E}(t) =& \Delta_B^{\mathrm{T}}(t) - \left(\Delta_B^{\mathrm{T}}(t)M(t) + M^{\mathrm{T}}(t)\Delta_B(t)\right)X(t) \\ & - \gamma F\left(E(t)\right) + \Delta_C(t). \end{aligned} \tag{17}$$

From the definition $E(t) = M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)$ and the condition that $\left(M^{\mathrm{T}}(t)M(t)\right)^{-1}$ exists for any time instant $t \in [0, +\infty)$, it follows that $X(t) = \left(M^{\mathrm{T}}(t)M(t)\right)^{-1}\left(E(t) + M^{\mathrm{T}}(t)\right)$. Substituting this equality into (17), we then get

$$\begin{aligned}\dot{E}(t) = &-\gamma F\left(E(t)\right) - \left(\Delta_B^{\mathrm{T}}(t)M(t) + M^{\mathrm{T}}(t)\Delta_B(t)\right)\\ &\cdot \left(M^{\mathrm{T}}(t)M(t)\right)^{-1}\left(E(t) + M^{\mathrm{T}}(t)\right)\\ &+ \left(\Delta_B^{\mathrm{T}}(t) + \Delta_C(t)\right).\end{aligned} \tag{18}$$

Let $B(t) := -\left(\Delta_B(t)^{\mathrm{T}}(t) + M^{\mathrm{T}}(t)\Delta_B(t)\right)\left(M^{\mathrm{T}}(t)M(t)\right)^{-1}$, equality (18) becomes

$$\dot{E}(t) = -\gamma F\left(E(t)\right) + B(t)E(t) + B(t)M^{\mathrm{T}}(t) + \Delta_B^{\mathrm{T}}(t) + \Delta_C(t),$$

which is equivalent to the following vector form:

$$\dot{e}(t) = -\gamma F\left(e(t)\right) + D(t)e(t) + D(t)a'(t) + b'(t) + c(t), \tag{19}$$

where $e(t) := \mathrm{vec}(E(t)) \in \mathbb{R}^{mn\times 1}$ represents a vector whose elements are obtained by piling all columns from $E(t)$ into a single column vector. The dimension of activation function array $F$ becomes $mn \times 1$ owing to the vectorization. Besides, $b'(t) := \mathrm{vec}(\Delta_B^{\mathrm{T}}(t)) \in \mathbb{R}^{mn\times 1}$, $c(t) := \mathrm{vec}(\Delta_C(t)) \in \mathbb{R}^{mn\times 1}$, $a'(t) = \mathrm{vec}(M^{\mathrm{T}}(t)) \in \mathbb{R}^{mn\times 1}$, and $D := I \otimes B(t)$ with symbol $\otimes$ denoting the Kronecker product, the detailed information about Kronecker product can be seen in [37,38].

Before further analyzing the robustness of proposed ZNN model (19), we firstly introduce an important lemma that will help us to complete the procedure.

**Lemma 2.** *Let $f_L(x)$, $f_S(x)$, $f_{PS}(x)$, $f_{Sbp}(x)$, $f_{SFTAF1}(x)$ and $f_{SFTAF2}(x)$ denote the linear (6), power-sigmoid (9), Sbp (10), SFTAF1 (11) and SFTAF2 (12) activation functions respectively. Then, these activation functions possess following properties.*

*(1) In the condition $|x| \in (0, 1)$, $|f_L(x)| < |f_{PS}(x)|$ always holds.*

*(2) In the condition $|x| \in (0, 1)$, $\beta_1 = \beta_2 = 1$, if $\tau \leqslant (2\xi\exp(-\xi))/(1-\exp(-2\xi))$, then $|f_{PS}(x)| \leqslant |f_{SFTAF1}(x)|$ and $|f_{PS}(x)| \leqslant |f_{SFTAF2}(x)|$ always hold, where the definition of $\xi$ and $\tau$ are the same as in function (9), (11) and (12).*

**Proof.**

(1) Define an auxiliary function $g(x) = \eta(1-\exp(-\xi x))/(1+\exp(-\xi x)) - x$, where $\eta = (1+\exp(-\xi))/(1-\exp(-\xi))$. Clearly, $g(x)$ satisfies $g(0) = g(1) = 0$. If we take the derivative of $g(x)$ with respect to $x$, we can get

$$g'(x) = \eta\frac{2\xi\exp(-\xi x)}{(1+\exp(-\xi x))^2} - 1.$$

Taking the derivative of $g'(x)$ with respect to $x$, we get

$$g''(x) = -2\eta\xi^2 e^{-\xi x}\frac{1-\exp(-2\xi x)}{(1+\exp(-\xi x))^4}.$$

Obviously, $g''(x) < 0$ if $x \in (0, 1)$. Thus $g(x)$ is a convex function in the domain field of $x \in (0, 1)$. Taking $g(0) = g(1) = 0$ into consideration, we conclude that if $x \in (0, 1)$, $g(x) > 0$. Since $\eta(1-\exp(-\xi x))/(1+\exp(-\xi x)) > 0$ and $x > 0$, $|f_L(x)| < |f_{PS}(x)|$ is true in $x \in (0, 1)$. Besides, $f_L(x)$, and $f_{PS}(x)$ are odd functions, so $|f_L(x)| < |f_{PS}(x)|$ holds true when $|x| \in (0, 1)$.

(2) Define auxiliary function $\psi(x) = \eta(1-\exp(-\xi x))/(1+\exp(-\xi x)) - \mathrm{sgn}^{\tau}(x)$ with $\xi > 2$, $0 < \tau < 1$, $\eta = (1+\exp(-\xi))/(1-\exp(-\xi))$, and $\mathrm{sgn}^{\tau}(x)$ formed by

$$\mathrm{sgn}^{\tau}(x) = \begin{cases} |x|^{\tau} & \text{if } x > 0,\\ 0 & \text{if } x = 0,\\ -|x|^{\tau} & \text{if } x < 0.\end{cases}$$

Clearly, $\psi(0) = \psi(1) = 0$. Firstly, let us consider the condition of $x \in (0, 1)$, and we can obtain the first order derivative of $\psi(x)$:

$$\psi'(x) = \eta\frac{2\xi\exp(-\xi x)}{(1+\exp(-\xi x))^2} - \tau x^{\tau-1}.$$

From $\psi'(x)$, we obtain $\lim_{x\to 0^+}\psi'(x) = -\infty$. Therefore, there must exist $\delta_1 > 0$, making $\psi'(x) > 0$ in $x \in (0, \delta_1)$. Furthermore, we get the second order derivative of $\psi(x)$:

$$\psi''(x) = -2\xi^2\eta\frac{1-\exp(-\xi x)}{(1+\exp(-\xi x))^4} + \tau(1-\tau)x^{\tau-2}.$$

Obviously, $\psi''(x)$ is a monotonically increasing function in the domain field of $x \in (0, 1)$. Because $\lim_{x\to 0^+}\psi''(x) = +\infty$, there must exist $\delta_2 > 0$ making $\psi''(x) > 0$ in the domain field of $x \in (0, \delta_2)$. Therefore, we should discuss the following two different conditions.

(1) If $\psi''(x) \geqslant 0$ is true when $x \in (0, 1)$, then we can find a point $a$ where $\psi'(x) < 0$ in $x \in (0, a)$, $\psi'(a) = 0$ and $\psi'(x) > 0$ in $x \in (a, 1)$. According to $\psi(x)$'s figure, $\psi(x) < 0$ holds for $x \in (0, 1)$.

(2) If $\psi''(x) < 0$ is true in interval $(b, 1)$, where $\psi''(b) = 0$. Then, $\psi'(x)$ will decrease in $(b, 1)$. Because $\psi'(1) = (2\xi\exp(-\xi))/(1-\exp(-2\xi)) - \tau \geqslant 0$, $\psi'(x)$ will not drop below zero in this case. Therefore, from the figure of $\psi(x)$ we know that $\psi(x) < 0$ still holds true in the domain of $x \in (0, 1)$.

From two conditions above, we can obtain $\psi(x) < 0$ in interval $(0, 1)$, which means that $f_{SFTAF1}(x) > f_{PS}(x) > 0$ when $x \in (0, 1)$. Since that $f_{SFTAF1}$ and $f_{PS}$ are odd functions, $|f_{NFTAF1}(x)| > |f_{PS}(x)|$ in $|x| \in (0, 1)$. Besides, we can easily obtain that $|f_{SFTAF2}(x)| > |f_{SFTAF1}(x)|$ when $|x| \in (0, 1)$, and $\beta_1 = \beta_2 = 1$, thus $|f_{SFTAF2}(x)| > |f_{PS}(x)|$.

The proof of Lemma 2 is now complete.

**Theorem 3.** *If $\|\Delta_B(t)\|_{\mathrm{F}} \leqslant \varepsilon_B$, $\|\Delta_C(t)\|_{\mathrm{F}} \leqslant \varepsilon_C$, and $\|M(t)\|_{\mathrm{F}} \leqslant \varepsilon_M$ hold true at any time point $t \geqslant 0$, with $0 < \varepsilon_B, \varepsilon_C, \varepsilon_M < +\infty$. Then in the condition of $\gamma > 2\varepsilon_A\varepsilon_B\varphi$, computation error $\|M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)\|_{\mathrm{F}}$ of the perturbed ZNN model (19) is limited under maximum value of $(\sqrt{l}+l)(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)/2(\gamma\rho - 2\varepsilon_M\varepsilon_B\varphi)$, where $l = mn$ is the total number of elements in matrix $M(t)$. Furthermore, the steady-state residual error $\|M^{\mathrm{T}}(t)M(t)X(t) - M^{\mathrm{T}}(t)\|_{\mathrm{F}}$ tends to decrease to zero as $\gamma \to +\infty$.*

**Proof.** Considering the vectorial perturbed ZNN model (19), we define a Lyapunov function candidate $v = \|e(t)\|_2^2/2 = e^{\mathrm{T}}(t)e(t)/2 = \sum_{i=1}^{nm} e_i^2(t)/2 \geqslant 0$. Clearly, $v$ is an positive definite variable and it satisfies that $v = 0$ only under the condition when $e(t) = 0$. Besides, $v$ has no upper bound if $\|e\|_2$ keeps increasing. Moreover, since $e^{\mathrm{T}}De$ is a scalar value, we get $e^{\mathrm{T}}De = (e^{\mathrm{T}}De)^{\mathrm{T}} = e^{\mathrm{T}}D^{\mathrm{T}}e$, which means $e^{\mathrm{T}}De = e^{\mathrm{T}}\frac{D+D^{\mathrm{T}}}{2}e$. Thus we can get the differential equation of $v$:

$$\begin{aligned}\dot{v} &= e^{\mathrm{T}}\dot{e} = e^{\mathrm{T}}(-\gamma F(e) + De + Da' + b' + c)\\ &= -\gamma e^{F}(e) + e^{\mathrm{T}}De + e^{\mathrm{T}}Da' + e^{\mathrm{T}}(b' + c)\\ &= -\gamma e^{\mathrm{T}}F(e) + e^{\mathrm{T}}\frac{D+D^{\mathrm{T}}}{2}e + e^{\mathrm{T}}Da' + e^{\mathrm{T}}(b' + c).\end{aligned} \tag{20}$$

For the first term of above equation, it satisfies that

$$-\gamma e^{\mathrm{T}}F(e) = -\sum_{i=1}^{l}\gamma|e_i|f(e_i). \tag{21}$$

Considering the second term of (20), based on the mathematical characteristic $\max_{1\leqslant r\leqslant n}|\lambda_r(B)|\leqslant\|B\|_{\mathrm{F}}$, we get

$$\begin{aligned} e^{\mathrm{T}}\frac{D+D^{\mathrm{T}}}{2}e &\leqslant e^{\mathrm{T}}e\max_{1\leqslant r\leqslant n^2}\left|\lambda_r\left(\frac{D+D^{\mathrm{T}}}{2}\right)\right| \\ &= e^{\mathrm{T}}e\max_{1\leqslant r\leqslant n^2}\left|\lambda_r\left(\frac{I\otimes B+(I\otimes B)^{\mathrm{T}}}{2}\right)\right| \\ &= e^{\mathrm{T}}e\max_{1\leqslant r\leqslant n^2}\left|\lambda_r\left(\frac{I\otimes (B+B^{\mathrm{T}})}{2}\right)\right| \\ &= e^{\mathrm{T}}e\max_{1\leqslant r\leqslant n^2}\left|\lambda_r\left(\frac{B+B^{\mathrm{T}}}{2}\right)\right| \\ &\leqslant e^{\mathrm{T}}e\|B\|_{\mathrm{F}}, \end{aligned} \tag{22}$$

where $\|B\|_{\mathrm{F}}$ satisfies following inequalities:

$$\begin{aligned} \|B\|_{\mathrm{F}} &= \left\|(\Delta_B^{\mathrm{T}}M+M^{\mathrm{T}}\Delta_B)(M^{\mathrm{T}}M)^{-1}\right\|_{\mathrm{F}} \\ &\leqslant \left(\left\|\Delta_B^{\mathrm{T}}M\right\|_{\mathrm{F}}+\left\|M^{\mathrm{T}}\Delta_B\right\|_{\mathrm{F}}\right)\left\|(M^{\mathrm{T}}M)^{-1}\right\|_{\mathrm{F}} \\ &= 2\|\Delta_B^{\mathrm{T}}M\|_{\mathrm{F}}\|(M^{\mathrm{T}}M)^{-1}\|_{\mathrm{F}} \\ &\leqslant 2\|\Delta_B^{\mathrm{T}}\|_{\mathrm{F}}\|M\|_{\mathrm{F}}\|(M^{\mathrm{T}}M)^{-1}\|_{\mathrm{F}} \\ &= 2\varepsilon_M\varepsilon_B\varphi. \end{aligned}$$

Thus, substituting above inequalities into (22), we obtain

$$e^{\mathrm{T}}\frac{D+D^{\mathrm{T}}}{2}e\leqslant e^{\mathrm{T}}e\|B\|_{\mathrm{F}}\leqslant e^{\mathrm{T}}e\cdot 2\varepsilon_M\varepsilon_B\varphi, \tag{23}$$

For the third term of (20), according to $Da'=\mathrm{vec}(BM^{\mathrm{T}})$, which is obtained from the Kronecker product's mathematical characteristics, and $\|\mathrm{vec}(BM^{\mathrm{T}})\|_2=\|BM^{\mathrm{T}}\|_{\mathrm{F}}$, we get

$$\begin{aligned} \|Da'\|_2 &= \left\|vec(BM^{\mathrm{T}})\right\|_2 \\ &= \left\|BM^{\mathrm{T}}\right\|_{\mathrm{F}} \\ &\leqslant \|B\|_{\mathrm{F}}\left\|M^{\mathrm{T}}\right\|_{\mathrm{F}} \\ &\leqslant 2\varepsilon_M^2\varepsilon_B\varphi. \end{aligned}$$

For the last term of (20), it follows from $\max_{1\leqslant i\leqslant l}|b'_i|\leqslant\|b'\|_2$ and $\max_{1\leqslant i\leqslant l}|c_i|\leqslant\|c\|_2$ that

$$\begin{aligned} e^{\mathrm{T}}(b'+c) &= e^{\mathrm{T}}b'+e^{\mathrm{T}}c \\ &\leqslant \sum_{i=1}^{l}|e_i|\max_{1\leqslant i\leqslant l}\left|[b']_i\right|+\sum_{i=1}^{l}|e_i|\max_{1\leqslant i\leqslant l}\left|[c]_i\right| \\ &\leqslant \sum_{i=1}^{l}|e_i|\|b'\|_2+\sum_{i=1}^{l}|e_i|\|c\|_2 \\ &\leqslant \sum_{i=1}^{l}|e_i|(\varepsilon_B+\varepsilon_C). \end{aligned} \tag{24}$$

Hence, in view of above inequalities (20), (21), (22), (23) and (24), we have

$$\begin{aligned} \dot v &\leqslant -\sum_{i=1}^{l}\big(\gamma|e_i|f(|e_i|)\big)+e^{\mathrm{T}}e\cdot 2\varepsilon_M\varepsilon_B\varphi \\ &\quad+\sum_{i=1}^{l}\big(|e_i|\cdot 2\varepsilon_M^2\varepsilon_B\varphi\big)+\sum_{i=1}^{l}\big(|e_i|(\varepsilon_B+\varepsilon_C)\big) \\ &= -\sum_{i=1}^{l}|e_i|\big(\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C\big). \end{aligned} \tag{25}$$

Having the above inequalities, we now analyze the following two situations.

(i) If $\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C\geqslant 0$ holds in time interval $[t_0,t_1)\forall i\in 1,2,\ldots,l$, then $\dot v\leqslant 0$ indicates that error vector $e(t)$ will always converge toward zero, which in the mean time implies that the state $X(t)$ of the perturbed ZNN model (19) converges toward the theoretical time-varying pseudoinverse of $M(t)$. If $\dot v=0$ is true when $t=t_2, t_2\in[0,+\infty)$, $e(t)$ reaches its steady state. Otherwise, the perturbed ZNN model (19) will fall into following circumstance along with $\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C$ decreasing.

(ii) Considering $\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C\leqslant 0$, $\exists i\in 1,2,\ldots,l$ at any time point $t$. The upper bound of $\dot v$ might be positive, implying $\dot v$ may be in two situations as $\dot v\leqslant 0$ and $\dot v>0$. In these cases, the error vector may not necessarily converge to zero, meaning the solution of the ZNN model may not converge to the theoretical solution $M^{+}(t)$. However, even in the worst case $\dot v>0$ when $e(t)$ diverges beyond, because $|e_i(t)|$ keeps increasing, the upper bound of $\dot v$ also decreases as long as $\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi>0$ is satisfied. In this case, as time evolves, there must exist a certain time instant $t_3$ making $\dot v(t_3)\leqslant 0$, thus $e(t)$ stays in steady state or decreases again.

Therefore, we obtain the conclusion that the residual error $\|e(t)\|$ will not always increase, and it is confined by a certain upper bound. Thus, inequality (25) can be rewritten as

$$\begin{aligned} \dot v &\leqslant -\sum_{i=1}^{l}|e_i|\big(\gamma f(|e_i|)-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C\big) \\ &= -\sum_{i=1}^{l}|e_i|\big(\gamma\rho_i|e_i|-2|e_i|\varepsilon_M\varepsilon_B\varphi-2\varepsilon_M^2\varepsilon_B\varphi-\varepsilon_B-\varepsilon_C\big), \end{aligned} \tag{26}$$

where there exists $\rho_i=f(|e_i|)/|e_i|\geqslant 1$. We now analyze a situation when any $|e_j|, j\in\{1,\ldots,l\}$ can achieve its highest value, when the upper bound of $\dot v(t)$ is zero in (26) according to the above analysis:

$$\begin{aligned} &\sum_{i\neq j}^{l}\big[(\gamma\rho_i-2\varepsilon_M\varepsilon_B\varphi)|e_i|^2-(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)|e_i|\big] \\ &+\big[(\gamma\rho_j-2\varepsilon_M\varepsilon_B\varphi)|e_j|^2-(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)|e_j|\big]=0. \end{aligned} \tag{27}$$

Evidently, the left side of the above equation (27) is a quadratic function of $|e_j|$. Besides, we can easily obtain:

$$\begin{aligned} &-\sum_{i\neq j}^{l}\big[(\gamma\rho_i-2\varepsilon_M\varepsilon_B\varphi)|e_i|^2-(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)|e_i|\big]\leq \\ &\sum_{i\neq j}^{l}\frac{(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)^2}{4(\gamma\rho_i-2\varepsilon_M\varepsilon_B\varphi)}\leq\frac{(l-1)(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)^2}{4(\gamma\rho-2\varepsilon_M\varepsilon_B\varphi)}, \end{aligned} \tag{28}$$

when $|e_i|=(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)/2(\gamma\rho_i-2\varepsilon_M\varepsilon_B\varphi)$, with parameter requirement $\gamma>2\varepsilon_M\varepsilon_B\varphi/\rho$, where $\rho=\min\{\rho_i|i=1,\ldots,l\}$. To calculate the upper bound of $|e_j|$, we refer to the characteristics of quadratic function about $|e_j|$, it yields from the above (27) and (28) that:

$$|e_j|\leq\frac{1}{2}(1+\sqrt{l})\frac{2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C}{\gamma\rho-2\varepsilon_M\varepsilon_B\varphi}.$$

Because $e_j$ can be any elements in $e(t)$, every element of $e(t)$ is bounded by the same formula:

$$\max_{1\leqslant i\leqslant l}|e_i(t)|\leqslant\frac{1}{2}(1+\sqrt{l})\frac{2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C}{\gamma\rho-2\varepsilon_M\varepsilon_B\varphi}. \tag{29}$$

If there exists any entry error $|e_k(t)|$, $k\in\{1,2,\ldots,l\}$ goes beyond inequality (29), $\dot v$ is limited to be negative, forcing $\|e(t)\|_2$ to decrease. In this situation, $|e_k(t)|$ decreases into bound to stop $v$ from decreasing. Therefore, inequality (29) indicates the absolute

value of all single element $|e_i(t)|,\ \forall i \in \{1,2,\ldots,l\}$ in $e(t)$ could not go beyond the upper bound. In addition, we obtain

$$\|E(t)\|_{\mathrm{F}} = \|e(t)\|_2 \leqslant \sqrt{\sum_{i=1}^{l} e_i^2(t)} \leqslant \sqrt{l}\max_{1\leqslant i\leqslant l}|e_i(t)|.$$

The above analysis leads to

$$\lim_{t\to+\infty}\|E(t)\|_{\mathrm{F}} \lesssim \frac{1}{2}(l+\sqrt{l})\frac{2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C}{\gamma\rho-2\varepsilon_M\varepsilon_B\varphi}.$$

The proof of Theorem 3 is now complete.

The proof of Theorem 3 not only proves that steady state residual error $\|M^{\mathrm{T}}(t)M(t)X(t)-M^{\mathrm{T}}(t)\|_{\mathrm{F}}$ is upper bounded around $(\sqrt{l}+l)(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)/2(\gamma\rho-2\varepsilon_M\varepsilon_B\varphi)$, but also proves that entry error $|e_i(t)|, \forall i \in \{1,2,\ldots,l\}$ is upper bounded around $(1+\sqrt{l})(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)/2(\gamma\rho-2\varepsilon_M\varepsilon_B\varphi)$. Evidently, the steady state residual error can be made very small by increasing design parameter $\gamma$. If $\gamma$ satisfies following inequalities

$$\gamma \geqslant (1+\sqrt{l})(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)+4\varepsilon_M\varepsilon_B\varphi/2\rho,$$

which is derived from $1 \geqslant ((1+\sqrt{l})(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C))/(2(\gamma\rho-2\varepsilon_M\varepsilon_B\varphi))$, then entry error $|e_i(t)|, \forall i \in \{1,2,\ldots,l\}$ would be bounded within interval [0, 1].

**Theorem 4.** *In addition to the general robustness result that obtained in Theorem 3, when $\gamma \geqslant \big((1+\sqrt{l})(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)+4\varepsilon_M\varepsilon_B\varphi\big)/2\rho$, the perturbed ZNN model (19) possesses following properties.*

*(1) If the power-sigmoid activation function $f_{PS}$ is used, then the upper bound of steady state residual error $\|E(t)\|_{\mathrm{F}}$ is smaller than that using linear activation function $f_L$.*

*(2) If activation function $f_{SFTAF1}$ is used with $\tau \leqslant (2\xi\exp(-\xi))/(1-\exp(-2\xi))$, then the upper bound of steady state residual error $\|E(t)\|_{\mathrm{F}}$ is smaller than using $f_{PS}$.*

*(3) If $\beta_1=\beta_2=1$, using activation function $f_{SFTAF2}$ can get even smaller steady-state residual error upper bound than that using $f_{SFTAF1}$.*

**Proof.** Obviously, when $\gamma \geqslant \big((1+\sqrt{l})(2\varepsilon_M{}^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)+4\varepsilon_M\varepsilon_B\varphi\big)/2\rho$, entry error $|e_i(t)|, \forall i \in \{1,2,\ldots,l\}$ falls into the interval [0, 1] as $t\to+\infty$. In this situation, the parts of activation functions defined in $x\in[-1,1]$ take effect. Theorem 3 implies that, for the perturbed ZNN model, we have $\|E(t)\|_{\mathrm{F}} \leqslant \frac{1}{2}(l+\sqrt{l})(2\varepsilon_M^2\varepsilon_B\varphi+\varepsilon_B+\varepsilon_C)/(\gamma\rho-2\varepsilon_M\varepsilon_B\varphi)$. Thus, under the same circumstances, the larger $\rho$ is, the smaller upper bound of $\|E(t)\|_{\mathrm{F}}$ will be. Besides, because all the activation functions $f(\cdot)$ are odd function, $f(x)<0$ when $x<0$. If $e_i(t)<0$, we have

$$\frac{|f(e_i(t))|}{|e_i(t)|} = \frac{-f(e_i(t))}{-e_i(t)} = \frac{f(-e_i(t))}{-e_i(t)} = \frac{f(|e_i(t)|)}{|e_i(t)|} = \rho.$$

For $e_i(t)>0$, evidently $\rho = |f(e_i(t))|/|e_i(t)|$ holds true too. Therefore, $\rho = |f(e_i(t))|/|e_i(t)|$ when $e_i(t)\neq 0$.

(1) Lemma 2 implies $|f_{PS}(x)| > |f_L(x)|,\ \forall |x|\in(0,1)$. Thus in the condition that $|e_i(t)|<1$, we obtain $\rho_{PS}/\rho_L = |f_{PS}(e_i(t))|/|f_l(e_i(t))| > 1$, which implies, by using power-sigmoid activation function, we are able to get smaller upper bound of steady-state error $\|E(t)\|_{\mathrm{F}}$ than that using linear activation function.

(2) Lemma 2 also proves that, in the condition of $\tau \leqslant (2\xi\exp(-\xi))/(1-\exp(-2\xi))$, $|f_{FTAF1}(x)| > |f_{PS}(x)|$ where $|x|\in(0,1)$. So, $\rho_{SFTAF1} > \rho_{PS}$ holds true when $|e_i(t)|<1$. Clearly, the upper bound of $\|E(t)\|_{\mathrm{F}}$ is smaller when using $f_{SFTAF1}$ than $f_{PS}$.

(3) In the condition that $\beta_1=\beta_2=1$, we can readily obtain $|f_{SFTFA2}(x)| > |f_{SFTAF1}(x)|$. Similarly, $\rho_{SFTAF2} > \rho_{SFTAF1}$, thus by using the SFTAF2, we get better steady-state residual upper bound than the SFTAF1.

The proof of Theorem 4 is now complete.

## 5. Numerical verification

To validate the superior convergence and robustness performance of the ZNN model (5) using our newly introduced SFTAF1 and SFTAF2, two numerical examples are introduced in this section. In addition, to illustrate the superiority of finite-time convergence compared to non-finite time convergence as well as the better robustness of the SFTAF1 and the SFTAF2, other activation functions including $f_L$, $f_{PS}$ and $f_{Sbp}$ are used in the numerical examples too.

### 5.1. Convergence discussion

Let us consider the following time-varying matrix $A(t)$ with full rank:

$$M(t) = \begin{bmatrix} \sin(3t) & \cos(3t) \\ -\cos(3t) & \sin(3t) \\ \sin(3t) & \cos(3t) \end{bmatrix} \in \mathbb{R}^{3\times 2}. \tag{30}$$

Then, according to Eq. (1), the theoretical time-varying pseudoinverse of matrix (30) can be obtained as

$$M^{+}(t) = \begin{bmatrix} 0.5\sin(3t) & -\cos(3t) & 0.5\sin(3t) \\ 0.5\cos(3t) & \sin(3t) & 0.5\cos(3t) \end{bmatrix}. \tag{31}$$

Since the theoretical pseudoinverse $M^{+}(t)$ is provided, we will use it as a criterion in the following convergence experiments to validate the superiority and efficacy of the SFTAF1 and the SFTAF2. The simulation results are shown in Figs. 1–3.

First, we investigate the finite-time convergent property of the SFTZNN1 and SFTZNN2 models. Without loss of generality, in this example, we set the parameter $\gamma=\beta_1=\beta_2=1$, and $\tau=0.2$. Just as shown in Fig. 1, beginning with a random selected initial state $X(0)\in\mathbb{R}^{2\times 3}$, state matrices $X(t)\in\mathbb{R}^{2\times 3}$ of SFTZNN1 and SFTZNN2 converge to the time-varying theoretical pseudoinverse (31) precisely and quickly. Besides, the residual error $\|E(t)\|_{\mathrm{F}} = \|M^{\mathrm{T}}(t)M(t)X(t)-M^{\mathrm{T}}(t)\|_{\mathrm{F}}$ is shown in Fig. 2. Note that, $X(0)\in\mathbb{R}^{2\times 3}$ is generated randomly from $[-1,1]\in\mathbb{R}^{2\times 3}$, and we can calculate the maximum initial entry error. We define the initial state matrix $X(0)$ as follows:

$$X(0) = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}.$$

Then, we obtain the initial error matrix $E(0)$ as follows:

$$\begin{aligned} E(0) &= M^{\mathrm{T}}(0)M(0)X(0) - M^{\mathrm{T}}(0) \\ &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}X(0) - \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 2x_{21}-1 & 2x_{22} & 2x_{23}-1 \\ x_{11} & x_{12}+1 & x_{13} \end{bmatrix}. \end{aligned}$$

So, we have the maximum initial entry error $|E_{\max}(0)| = \max\big\{|E^{+}(0)|, |E^{-}(0)|\big\} \leqslant 3$. According to Theorem 1, we can obtain the convergence time upper bounds of SFTZNN1 and SFTZNN2 as

$$t_1 = \frac{3^{(1-0.2)}}{1-0.2} \approx 3.01\ \mathrm{s},$$

$$t_2 = \frac{1}{0.8}\ln\Big(\frac{3^{0.8}+1}{1}\Big) \approx 1.5327\ \mathrm{s}.$$

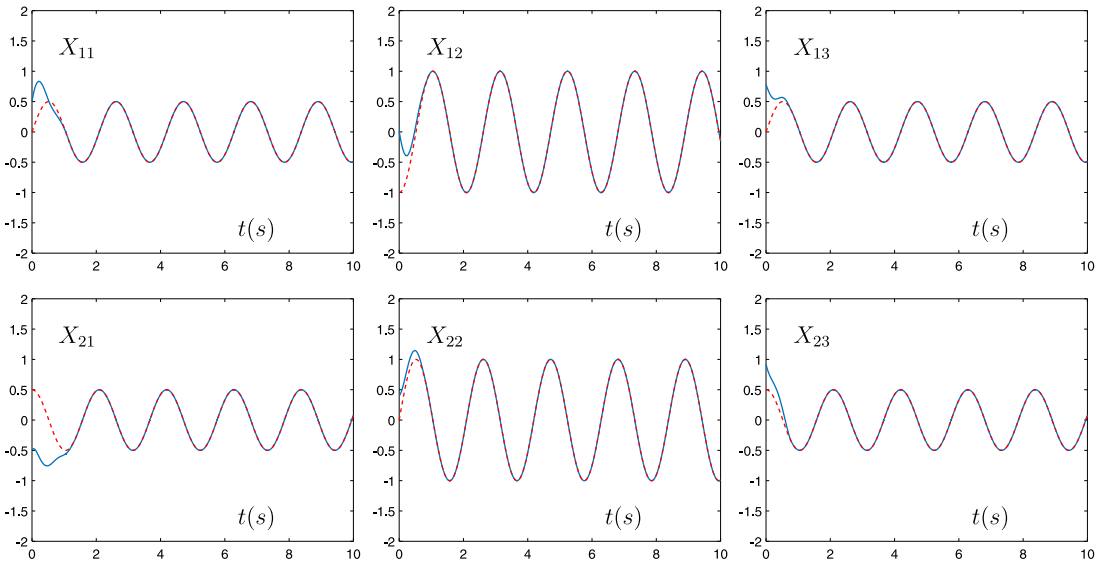


**Fig. 1.** Solutions obtained by ZNN model (5) activated by SFTAF2 (12), where red dotted curves denote theoretical pseudoinverse of (30) and blue solid curves denote the solution.

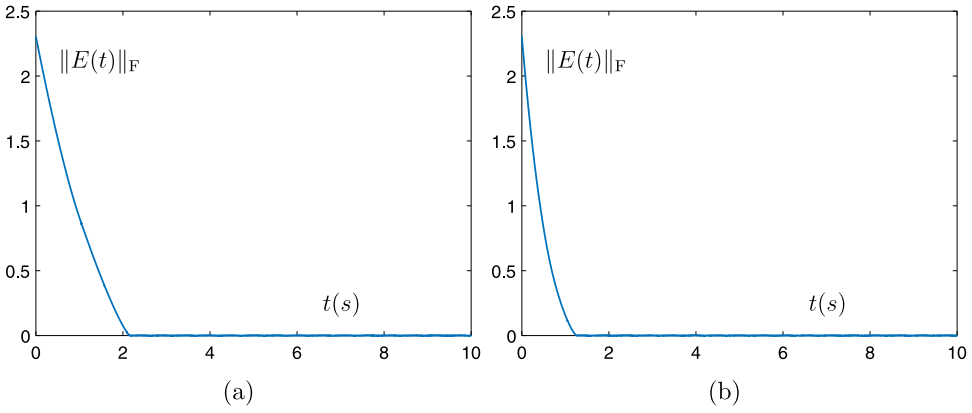


**Fig. 2.** Steady-state error $\|E(t)\|_{\mathrm{F}}$ produced by ZNN model (5) for the time-varying pseudoinverse.(a) Activated by SFTAF1 with $\tau = 0.2$. (b) Activated by SFTAF2 with $\tau = 0.2$.

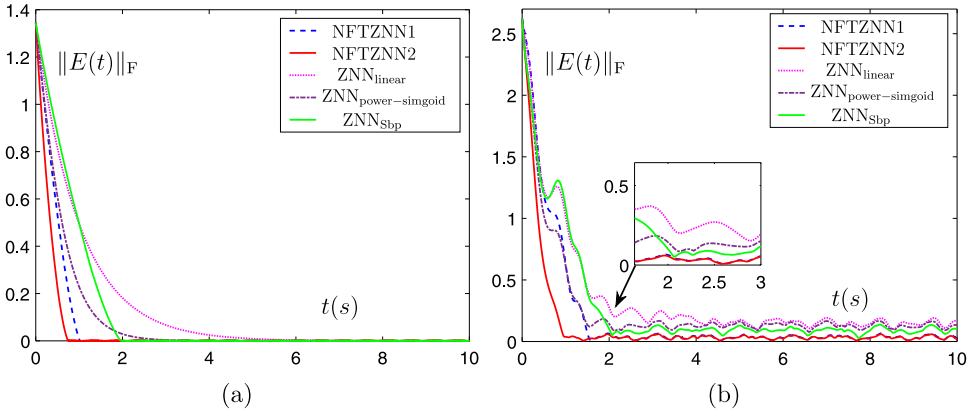


**Fig. 3.** Trajectories of residual error $\|E(t)\|_{\mathrm{F}}$.

Clearly, as shown in Fig. 2, the convergence time of SFTZNN1 and SFTZNN2 both fall within theoretical upper-bounds, which verifies the correctness of Theorem 1 in the experimental aspect.

Then we investigate the convergence speed superiority of the SFTAF1 and the SFTAF2 when compared to other aforementioned activation functions. In this case, we set the design parameter $p = 3$ and $\xi = 4$ in $f_{\mathrm{PS}}$. For $f_{\mathrm{SFTAF1}}$, $f_{\mathrm{SFTAF2}}$ and $f_{\mathrm{Sbp}}$, we set $\tau = 0.2$, and $\beta_1 = \beta_2 = 1$. The transient trajectories of residual errors $\|E(t)\|_{\mathrm{F}}$ synthesized by ZNN model (5) using different activation functions are depicted in Fig. 3(a). Evidently, Fig. 3(a) shows that $\|E(t)\|_{\mathrm{F}}$ synthesized by $f_L$, and $f_{\mathrm{PS}}$ converge exponentially to zero, instead of diminish directly to zero like $f_{\mathrm{NFTAF1}}$, $f_{\mathrm{SFTAF2}}$

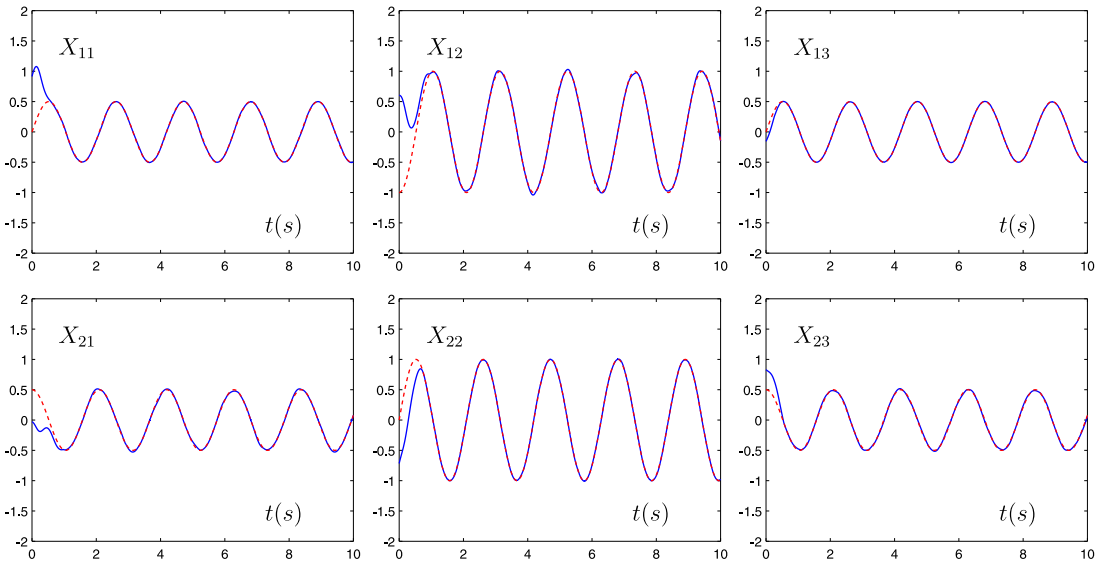


**Fig. 4.** State solutions of the perturbed ZNN model (19) activated by SFTAF2 (12), where red dotted curves represent the theoretical solution (31) and blue solid curves represent the solution obtained by neural network.

and $f_{\text{Sbp}}$. This finding verifies the correctness of Proposition 2 in the experimental aspect. More importantly, Fig. 3(a) illustrates that $f_{\text{SFTAF1}}$ and $f_{\text{SFTAF2}}$ possess better convergence speed compared to other activation functions, where residual error of SFTZNN2 converges to zero within 1.6 s, which is even better than that of SFTZNN1 (about 3 s).

### 5.2. Robustness discussion

For the purpose of showing robustness superiority of our newly introduced activation function SFTAF1 (11) and SFTAF2 (12), we evaluate the performance of the perturbed ZNN model (19) with two types of perturbations, which are described as the following form

$$\Delta_B(t) = \varepsilon_1 \begin{bmatrix} \sin(8t) & -\sin(8t) \\ \cos(8t) & \sin(8t) \\ -\cos(8t) & \cos(8t) \end{bmatrix},$$

$$\Delta_C(t) = \varepsilon_2 \begin{bmatrix} 0 & -\sin(8t) & 0 \\ \sin(8t) & 0 & \cos(8t) \end{bmatrix},$$

with $\varepsilon_1 = \varepsilon_2 = 0.5$. The design parameters of aforementioned activation functions are set to be the same as previous one.

As can be observed from Fig. 4 which is produced under the large differentiation error and implementation error, the perturbed ZNN model (19) activated by SFTAF2 still converges to the theoretical pseudoinverse with small residual error. Besides, the state solution trajectories of SFTAF1 are similar, and thus are admitted here. Fig. 3(b) shows that, when using the SFTAF1 or SFTAF2 activation function, the steady-state residual error is always smaller than that using activation functions (6), (9) and (10). Fig. 3(b) also shows that, compared with the SFTAF1 function, SFTAF2 achieves better result with faster convergence rate under perturbation. These findings inversely verifies the correctness of Theorem 4.

Moreover, we observed from simulation data that the upper bound of residual error decreases when the design parameter increases. In simulation, magnitude of the maximum steady-state residual error becomes about 10 times and 100 times smaller than that of $\gamma = 1$ when we set $\gamma = 10$ and $\gamma = 100$ respectively. This is exactly what we expect according to Theorem 3.

## 6. Application on manipulator kinematic control

In this section, we will apply the proposed neural dynamics to control a three-link robot by obtaining the time-varying pseudoinverse matrix.

### 6.1. Introduction to kinematics control

Given a redundant robot manipulator, its end-effector's three-dimensional coordinate in space is denoted by a vector $r(t) \in \mathbb{R}^m$, and its joint-space vector is denoted by $\theta(t) \in \mathbb{R}^n$. The relationship of these two vectors can be described by the following mathematical model [27,39]:

$$r(t) = f(\theta(t)), \tag{32}$$

where $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ denotes a matrix function mapping from $\theta(t)$ to $r(t)$. The above equation is determined by the mechanical structure of the specified robot manipulator. Then, the inverse kinematics problem is defined as finding the control variables $\theta(t)$ corresponding to any given trajectories $r(t)$. Unfortunately, due to nonlinear property of $f(\cdot)$, it is usually impossible for us to find analytical solution of $f^{-1}$. Therefore we cannot get the analytical solution of $\theta(t)$ through $f^{-1}(r(t))$.

Thus, the inverse kinematics problem is generally solved at the velocity level. Differentiating (32) with respect to $t$, we obtain the relationship between velocity $\dot{r}(t)$ and $\dot{\theta}(t)$:

$$J(\theta(t))\dot{\theta}(t) = \dot{r}(t), \tag{33}$$

where $J(\theta) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix defined as $J(\theta) = \partial f(\theta)/\partial\theta$ and $\dot{\theta}(t) \in \mathbb{R}^n$ is the joint velocity. Because $m < n$ in the redundant manipulator, (33) obviously have infinite number of solutions.

One solution to (33) is to use the pseudoinverse of $J(\theta)$ which is widely adopted by current research. Generally, pseudoinverse type solution is formulated as minimum velocity norm (MVN) form [28]:

$$\dot{\theta}(t) = J^{+}(t)(\dot{r}(t) + \lambda_b(r(t) - f(\theta))),$$

where $J^{+}(t) = J^{\mathrm{T}}(t)(J(t)J^{\mathrm{T}}(t))^{-1} \in \mathbb{R}^{n\times m}$ denotes the right Moore–Penrose inverse of $J(t)$, and $\lambda_b > 0 \in \mathbb{R}$ is the feedback gain used to ensure the precision of the end-effector position. Now,

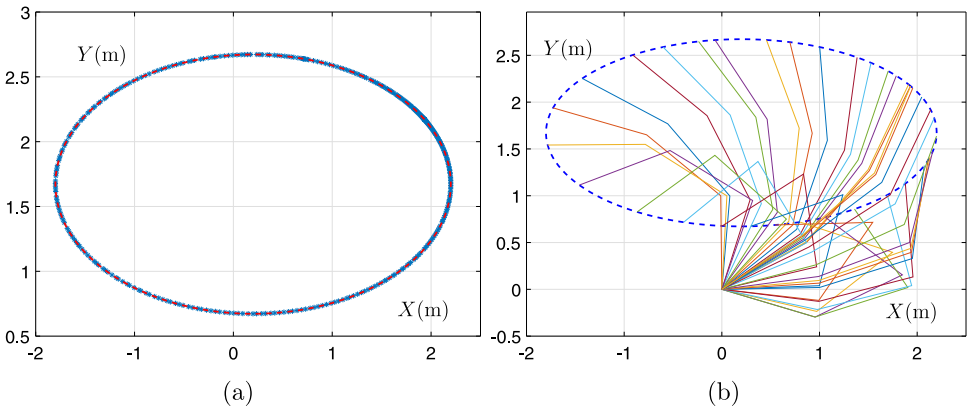


**Fig. 5.** Positioning result and top view of ellipse-path tracking task using the SFTZNN2. (a) Difference between the ellipse path and the actual end-effector position. (b) Top view of the whole tracking motion process.

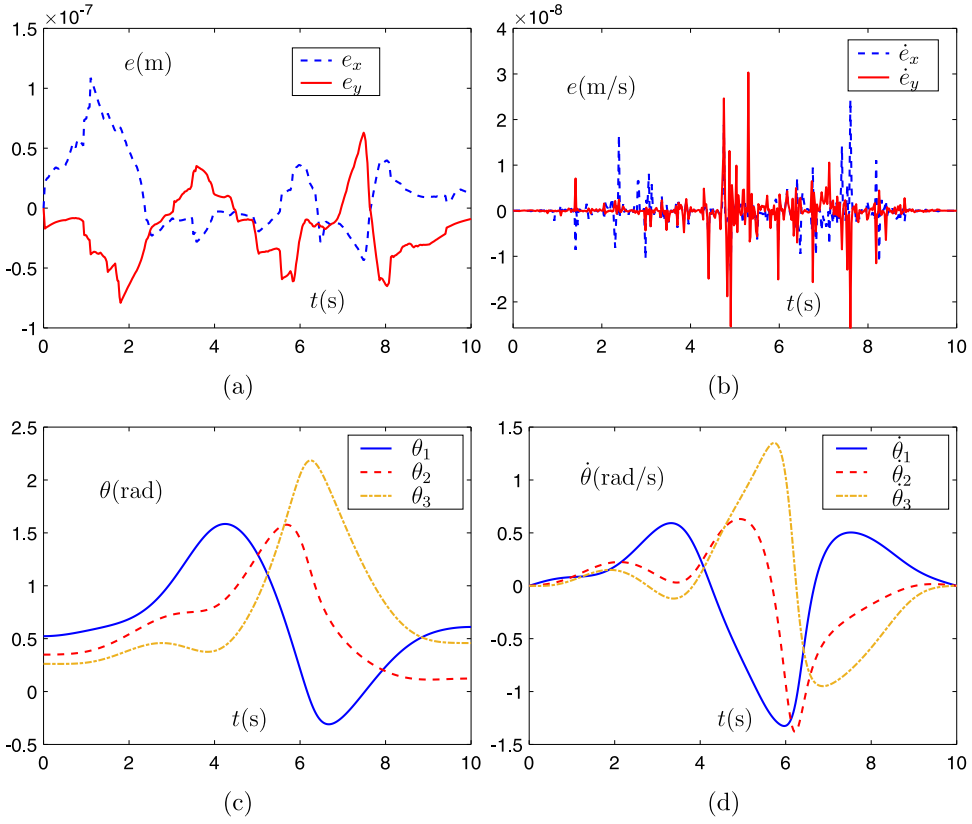


**Fig. 6.** End-effector and joint parameters together with some of their errors when accomplishing the ellipse-path tracking task. (a) Position error. (b) Joint velocity error. (c) Joint angle. and (d) Joint velocity.

we apply our ZNN model (5) to such a MVN scheme. That is, we use the following dynamics to obtain $J^{+}(t)$:

$$\dot{X}(t)J(t)J^{\mathrm{T}}(t)+X(t)\big(\dot{J}(t)J^{\mathrm{T}}(t)+J(t)\dot{J}^{\mathrm{T}}(t)\big) =\dot{J}^{\mathrm{T}}(t)-\gamma F\big(X(t)J(t)J^{\mathrm{T}}(t)-J^{\mathrm{T}}(t)\big).$$

### 6.2. Simulation based on three-link planar manipulator

According to the mechanical structure of the three-link planar robot manipulator, its Jacobian matrix $J(\theta)$ can be obtained as follows:

$$J=\begin{bmatrix} -s_1-s_2-s_3 & -s_2-s_3 & -s_3 \\ c_1+c_2+c_3 & c_2+c_3 & c_3 \end{bmatrix},$$

where we set the length of each link $l_i, i=1,2,3$ as 1 m, with $s_i=\sin(\sum_{j=1}^{i}\theta_j)$ and $c_i=\cos(\sum_{j=1}^{i}\theta_j)$. In addition, we set the design parameter $\gamma=1$, and $\lambda_b=1$. The duration is set as 10 s, and the initial joint angle being set as $\theta(0)=[\pi/6,\pi/9,\pi/12]$.

The desired path of the end-effector is an ellipse path and the simulation results are shown in Figs. 5–6. Fig. 5 illustrates that the end-effector tracks the expected ellipse-path sufficiently close. Fig. 6(a) and (b) shows that the positioning/velocity errors between the actual path and the expected path is at the order of $10^{-7}$ m and $10^{-8}$ m. Such a magnitude is clearly usable in real world applications. Besides, it can be observed from Fig. 6(c) and (d) that the change of joint angle $\theta(t)$ and joint velocity $\dot{\theta}(t)$ is smooth, which indicates our ZNN model (5) is very suitable for such engineering applications.

The simulation results of controlling a planar manipulator using ZNN model (5) substantiates the superiority of SFTAF1 and SFTAF2. These observations also validate the effectiveness of this neural network with activation function SFTAF1 and SFTAF2 when applied to control redundant manipulator.

## 7. Conclusion

Two simplified finite-time activation functions which called the SFTAF1 and the SFTAF2 are exploited to modify zero neural network (ZNN) for solving the pseudoinverse of time-varying matrix in real time. Different from traditional activation functions like the linear function and the power-sigmoid function, SFTAF1 and SFTAF2 make ZNN achieve finite-time convergence rather than the exponential convergence. That is, the SFTAF1 and SFTAF2 activated ZNN models (called the SFTZNN1 and the SFTZNN2) are able to obtain the exact solution of the time-varying matrix pseudoinverse within an certain amount of time. Furthermore, it has been theoretically validated that the SFTZNN1 and SFTZNN2 models possess superior convergence and robustness performance, with simpler structure when compared with the Sbp function. The numerical verifications demonstrate the theoretical analysis, high accuracy and superiority of the SFTZNN1 and SFTZNN2 models. In addition, a successful application of the SFTZNN2 model to control a three-link planar is also presented. Future work will be accelerating the SFTZNN1 and SFTZNN2 models to fixed-time convergence when solving time-varying matrix pseudoinverse, and further enhance robustness of such two ZNN models in presence of different types of noises.

## CRediT authorship contribution statement

**Zeshan Hu:** Data curation, Writing - original draft, Visualization. **Lin Xiao:** Conceptualization, Methodology, Supervision, Investigation, Writing - review & editing. **Kenli Li:** Supervision. **Keqin Li:** Validation. **Jichun Li:** Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61866013, Grant 62066015, Grant 61976089, Grant 61966014, and Grant 61563017 and in part by the Natural Science Foundation of Hunan Province of China under Grant 2019JJ50478, Grant 18A289, Grant 2018TP1018, and Grant 2018RS3065.

## References

[1] G. Plonka, S. Hoffmann, J. Weickert, Pseudo-inverses of difference matrices and their application to sparse signal approximation, Linear Algebra Appl. 503 (2016) 26–47.
[2] S. Devadithya, A. Pedross-Engel, C. Watts, M. Reynolds, Partitioned inverse image reconstruction for millimeter-wave SAR imaging, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2017, pp. 6060–6064.
[3] M. Liang, B. Zheng, Further results on Moore–Penrose inverses of tensors with application to tensor nearness problems, Comput. Math. Appl. 77 (5) (2019) 1282–1293.
[4] X. Chen, H. Zhao, H. Sun, S. Zhen, Adaptive robust control based on Moore-Penrose generalized inverse for underactuated mechanical systems, IEEE Access 7 (2019) 157136–157144.
[5] L. Jin, S. Li, H. Wang, Z. Zhang, Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence, Appl. Soft. Comput. 62 (2018) 840–850.
[6] Y. Wei, J. Cai, M. Ng, Computing Moore-Penrose inverses of Toeplitz matrices by Newton's iteration, Math. Comput. Modelling 40 (1–2) (2004) 181–191.
[7] F. Huang, X. Zhang, An improved Newton iteration for the weighted Moore–Penrose inverse, Appl. Math. Comput. 174 (2) (2006) 1460–1486.
[8] J. Zhou, Y. Zhu, X. Li, Z. You, Variants of the Greville formula with applications to exact recursive least squares, SIAM J. Matrix Anal. Appl. 24 (1) (2002) 150–164.
[9] H. Wang, J. Li, H. Liu, Practical limitations of an algorithm for the singular value decomposition as applied to redundant manipulators, in: 2006 IEEE Conference on Robotics, Automation and Mechatronics, IEEE, 2006, pp. 1–6.
[10] H. Han, Y. Li, Y. Guo, J. Qiao, A soft computing method to predict sludge volume index based on a recurrent self-organizing neural network, Appl. Soft Comput. 38 (2016) 477–486.
[11] F.F. El-Sousy, Adaptive hybrid control system using a recurrent RBFN-based self-evolving fuzzy-neural-network for PMSM servo drives, Appl. Soft Comput. 21 (2014) 509–532.
[12] L. Liu, Z. Wang, H. Zhang, Neural-network-based robust optimal tracking control for MIMO discrete-time systems with unknown uncertainty using adaptive critic design, IEEE Trans. Neural Netw. Learn Syst. 29 (4) (2018) 1239–1251.
[13] A. Hosseini, J. Wang, S. Hosseini, A recurrent neural network for solving a class of generalized convex optimization problems, Neural Netw. 44 (2013) 78–86.
[14] L. Xiao, Y. Zhang, K. Li, B. Liao, Z. Tan, A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion, Neurocomputing 331 (2019) 483–492.
[15] J.W. S. Qin, A neurodynamic optimization approach to bilevel quadratic programming, IEEE Trans. Neural Netw. Learn. Syst. 28 (11) (2017) 2580–2591.
[16] L. Jin, S. Li, B. Hu, M. Liu, A survey on projection neural networks and their applications, Appl. Soft Comput. 76 (2019) 533–544.
[17] D. Chen, S. Li, L. Liao, A recurrent neural network applied to optimal motion control of mobile robots with physical constraints, Appl. Soft Comput. 85 (2019) 105880.
[18] L. Xiao, K. Li, Z. Tan, Z. Zhang, B. Liao, K. Chen, L. Jin, S. Li, Nonlinear gradient neural network for solving system of linear equations, Inf. Process. Lett. 142 (2019) 35–40.
[19] P. Stanimirović, M. Petković, D. Gerontitis, Gradient neural network with nonlinear activation for computing inner inverses and the Drazin inverse, Neural Process. Lett. 48 (1) (2018) 109–133.
[20] Y. Zhang, S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, IEEE Trans. Neural Netw. Learn. Syst. 16 (6) (2005) 1477–1490.
[21] W. Li, A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations, IEEE Trans. Industr. Inform. 14 (12) (2018) 5289–5298.
[22] S. Li, S. Chen, B. Liu, Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function, Neural Process. Lett. 37 (2) (2013) 189–205.
[23] S. Li, Y. Li, Z. Wang, A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application, Neural Netw. 39 (2013) 27–39.
[24] L. Xiao, K. Li, M. Duan, Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: A unified framework for zeroing neural network, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3360–3369.
[25] L. Xiao, B. Liao, S. Li, K. Chen, Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations, Neural Netw. 98 (2018) 102–113.
[26] Y. Shen, P. Miao, Y. Huang, Y. Shen, Finite-time stability and its application for solving time-varying Sylvester equation by recurrent neural network, Neural Process. Lett. 42 (3) (2015) 763–784.
[27] X. Lv, Z. Tan, K. Chen, Z. Yang, Improved recurrent neural networks for online solution of Moore-Penrose inverse applied to redundant manipulator kinematic control, Asian J. Control (2018) 1–9.
[28] D. Guo, Y. Zhang, Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion, Appl. Soft Comput. 24 (2014) 158–168.
[29] Q. Xiang, B. Liao, L. Jin, M. Liu, Y. Zhang, Nonlinearly activated neural network for solving dynamic complex-valued matrix pseudoinverse, in: 2017 36th Chinese Control Conference, CCC, IEEE, 2017, pp. 3888–3892.
[30] Q. Xiang, B. Liao, L. Xiao, L. Jin, A noise-tolerant Z-type neural network for time-dependent pseudoinverse matrices, Optik 165 (2018) 16–28.

[31] P. Miao, Y. Shen, X. Xia, Finite time dual neural networks with a tunable activation function for solving quadratic programming problems and its application, Neurocomputing 143 (2014) 80–89.
[32] J. Barata, M. Hussein, The Moore–Penrose pseudoinverse: A tutorial review of the theory, Braz. J. Phys. 42 (1–2) (2012) 146–165.
[33] B. Liao, Y. Zhang, From different ZFs to different ZNN models accelerated via Li activation functions to finite-time convergence for time-varying matrix pseudoinversion, Neurocomputing 133 (2014) 512–522.
[34] L. Xiao, Z. Zhang, S. Li, Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators, IEEE Trans. Syst. Man. Cybern. Syst. 49 (11) (2018) 2210–2220.
[35] P. Miao, Y. Shen, Y. Li, L. Bao, Finite-time recurrent neural networks for solving nonlinear optimization problems and their application, Neurocomputing 177 (2016) 120–129.
[36] C. Mead, M. Ismail, Analog VLSI Implementation of Neural Systems, Vol. 80, Springer Science & Business Media, 2012.
[37] Z. Yang, R. Huang, W. Zhu, J. Liu, Accurate solutions of structured generalized Kronecker product linear systems, Numer. Algorithms (2020) 1–22.
[38] J. Chen, S. Xu, B. Zhang, Z. Qi, Z. Li, Novel stability conditions for discrete-time T–S fuzzy systems: a Kronecker-product approach, Inform. Sci. 337 (2016) 72–81.
[39] P. Miao, Y. Shen, Y. Huang, Y. Wang, Solving time-varying quadratic programs based on finite-time Zhang neural networks and their application to robot tracking, Neural Comput. Appl. 26 (3) (2015) 693–703.